

Appl. No. 10/737,418

Atty. Dkt. No. NVDA/P001024

Reply to Final Office Action of August 31, 2005

**IN THE CLAIMS:**

Please amend the claims as follows:

1. (Currently Amended) A method of avoiding a position conflict during fragment shading, comprising:

partitioning an image into tiles, each tile associated with a unique tile origin;

allocating an entry in at least one storage resource to a tile origin, the entry specified by a tile number; and

storing a portion of a position associated with at least one fragment in the entry, wherein the at least one fragment is produced by a graphics primitive intersecting a pixel within a tile identified by the tile number; and

delaying processing of any fragment within the tile that corresponds to the portion of the position stored in the entry.

2. (Original) The method of claim 1, further comprising storing a coverage mask associated with the at least one fragment in the entry.

3. (Original) The method of claim 1, further comprising storing a tile state in the entry.

4. (Original) The method of claim 1, further comprising storing a timestamp value in the entry.

5. (Original) The method of claim 2, further comprising:

combining the coverage mask associated with the at least one fragment with coverage mask data associated with at least one other fragment to produce combined coverage mask data; and

storing the combined coverage mask data in the entry.

Page 2

398178\_1

Appl. No. 10/737,418

Any. Dkt. No. NVDA/P001024

Reply to Final Office Action of August 31, 2005

6. (Previously Presented) The method of claim 1, wherein the number of entries in the at least one storage resource is less than the number of tiles in the image.
7. (Original) The method of claim 1, further comprising outputting the position associated with at least one fragment when a position conflict does not exist.
8. (Original) The method of claim 1, further comprising outputting a token when a predetermined number of quads are received.
9. (Original) The method of claim 8, further comprising updating a timestamp when the token is output.
10. (Previously Presented) The method of claim 1, outputting a token when a position conflict is detected within a tile and delaying processing of a fragment that corresponds to the position conflict.
11. (Previously Presented) A conflict detection unit configured to detect position conflicts during fragment shading, comprising:
- at least one storage resource including entries configured to store tile origins and coverage mask data; and
- a control unit configured to determine whether or not a position conflict exists during fragment shading within a tile associated with a tile origin stored in the at least one storage resource and output a stall signal indicating whether or not the conflict detection unit will accept new data.
12. (Original) The conflict detection unit of claim 11, wherein the control unit is configured to output a token when a position conflict is detected.

Page 3

398178\_1

Appl. No. 10/737,418

Atty. Dkt. No. NVDA/P001024

Reply to Final Office Action of August 31, 2005

13. (Original) The conflict detection unit of claim 11, wherein the control unit is configured to output a token when all of the entries within the at least one storage resource are allocated.
14. (Original) The conflict detection unit of claim 11, wherein the control unit is configured to output a token when a predetermined number of quads are received.
15. (Original) The conflict detection unit of claim 11, wherein the control unit is configured to combine a coverage mask with coverage mask data read from an entry in the at least one storage resource to produce combined coverage mask data and write the combined coverage mask data to the entry.
16. (Original) The conflict detection unit of claim 11, further comprising a timestamp unit configured to maintain a timestamp.
17. (Original) The conflict detection unit of claim 16, wherein the control unit is configured to copy the timestamp from the timestamp unit to an entry in the at least one storage resource.
18. (Previously Presented) The conflict detection unit of claim 11, further comprising a read interface configured to read data from at least one buffer.
19. (Previously Presented) The conflict detection unit of claim 11, further comprising a fragment processing unit configured to receive fragments produced by rasterizing graphics primitives, each fragment associated with a position, and the data read from the at least one buffer and generate processed fragments.
20. (Original) The conflict detection unit of claim 19, further comprising a write interface configured to write the processed fragments to the one or more buffers.

Page 4

398178\_1